Package ‘L2DensityGoFtest’

March 25, 2020

Encoding UTF-8

Type Package

Title Density Goodness-of-Fit Test

Version 0.1.0

Author Dimitrios Bagkavos [aut, cre]

Maintainer Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

Description Provides functions for the implementation of a density goodness-of-fit test, based on piecewise approximation of the L2 distance.

Imports fGarch, kedd, nor1mix

License GPL (>= 2)

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-25 15:30:16 UTC

R topics documented:

hopt.cutoff — *Power-optimal bandwidth and critical value*

Description

Implements the power-optimal bandwidth for density goodness-of-fit testing and finite sample critical value based on Edgwworth expansions.

Usage

```
hopt.cutoff(xin, dist, kfun, p1, p2, sig.lev)
```

Arguments

`xin`	A vector of data points - the available sample size.
`dist`	The null distribution.
`kfun`	The kernel to use in the density estimates used in the bandwidth expression.
`p1`	Argument 1 (vector or object) for the null distribution.
`p2`	Argument 2 (vector or object) for the null distribution.
`sig.lev`	Significance level of the hypothesis test.

Details

Implements:

1. The power-optimal bandwidth for the test statistic `S.n` given by

$$h = \{\frac{\sqrt{2}K^{(3)}(0)}{3R(K)^{3/2}}\frac{\nu_2}{R(f)^{3/2}}\}^{-1/2}\{\frac{n\int\Delta_n^2(x)f^2(x)\,dx}{\sigma^2\{2\nu_2 R(K)\}^{1/2}}\}^{-3/2}$$

2. Finite sample critical value based on Edgwworth expansions, given by

$$l_\alpha = z_\alpha + d_0\sqrt{h} + d_2(n\sqrt{h})^{-1}$$

   where z_α is the α quantile of the normal distribution.

Both quantities are similar in nature to the corresponding optimal bandwidth and critical value estimates obtained for the colsely relatated regression setting in Gao and Gijbels (2008).

Value

A vector containing the estimates of two values: power-optimal bandwidth and critical value at the given significance level

Author(s)

Dimitrios Bagkavos

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

References

Gao and Gijbels, Bandwidth selection in nonparametric kernel testing, pp. 1584-1594, JASA (2008)

---

kde *Kernel Density Estimation*

Description

Implements the (classical) kernel density estimator, see (2.2a) in Silverman (1986).

Usage

```
kde(xin, xout, h, kfun)
```

Arguments

`xin`	A vector of data points. Missing values not allowed.
`xout`	A vector of grid points at which the estimate will be calculated.
`h`	A scalar, the bandwidth to use in the estimate, e.g. `bw.nrd(xin)`
`kfun`	Kernel function to use. Supported kernels: `Epanechnikov`, `Biweight`, `Gaussian`, `Rectangular`, `Triangular`.

Details

The classical kernel density estimator is given by

$$\hat{f}(x;h) = n^{-1}\sum_{i=1}^{n} K_h(x - X_i)$$

h is determined by a bandwidth selector such as Silverman's default plug-in rule.

Value

A vector with the density estimates at the designated points xout.

Author(s)

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

References

Silverman (1986), Density Estimation for Statistics and Data Analysis, Chapman and Hall, London.

Examples

```
x<-seq(-5, 5,length=100)          #design points where the estimate will be calculated
plot(x, dnorm(x),  type="l", xlab = "x", ylab="density") #plot true density function
SampleSize <- 100
ti<- rnorm(SampleSize)            #draw a random sample from the actual distribution

huse<-bw.nrd(ti)
arg2<-kde(ti, x, huse, Epanechnikov) #Calculate the estimate
lines(x, arg2, lty=2)             #draw the result on the graphics device.
```

Kernels *Kernel functions*

Description

Implements various kernel functions, including boundary, integrated and discrete kernels for use in the definition of the nonparametric estimates

Usage

```
Biweight(x, ...)
Epanechnikov(x, ...)
Triangular(x, ...)
Gaussian(x, ...)
Rectangular(x, ...)
```

Arguments

x	A vector of data points where the kernel will be evaluated.
...	Further arguments.

Details

Implements the Biweight, Triangular, Guassian, Rectangular and Epanechnikov kernels.

Value

The value of the kernel at x

References

1. Bagkavos and Patil, Local Polynomial Fitting in Failure Rate Estimation, IEEE Transactions on Reliability, 57, (2008),
2. Bagkavos (2011), Annals of the Institute of Statistical Mathematics, 63(5), 1019-1046,

NDistDens — *Select null distribution*

Description

Implements the selection of null distribution; to be used within the implementation of the test statistic S.n

Usage

```
NDistDens(x, dist, p1, p2)
```

Arguments

`x`	A vector of data points - the available sample size.
`dist`	The null distribution.
`p1`	Argument 1 (vector or object) for the null distribution.
`p2`	Argument 2 (vector or object) for the null distribution.

Details

Implements the null distribution evaluation at designated points, given the parameters p1 and p2.

Value

A vector containing the density values of the designated distribution

Author(s)

Dimitrios Bagkavos

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

References

Will be added in due course

`S.n` *Goodness-of-Fit test statistic based on discretized L2 distance*

Description

Implements a density goodness of fit test based on a discretized approximation of the L2 distance.

Usage

```
S.n(xin, h, drate, dist, p1, p2)
```

Arguments

`xin`	A vector of data points - the available sample size.
`h`	The bandwidth to use, typically the output of `hopt.cutoff`.
`drate`	power to use for the binning, the default is 3/4.
`dist`	The null distribution.
`p1`	Argument 1 (vector or object) for the null distribution.
`p2`	Argument 2 (vector or object) for the null distribution.

Details

Implements the test statistic used for testing the hypothesis

$$H_0 : f(x) = f_0(x, p1, p2) vs H_a : f(x) \neq f_0(x, p1, p2).$$

This density goodness-of-fit test is based on a discretized approximation of the L2 distance. Assuming that n is the number of observations and $g = (max(xin) - min(xin))/n^{-drate}$ is the number of bins in which the range of the data is split, the test statistic is:

$$S_n(h) \frac{1}{g(g-1)h} \sum \sum_{i \neq j} K\{(x_i - x_j)h^{-1}\}\{Y_i - f_0(x_i)\}\{Y_j - f_0(x_j)\}$$

where K is the Epanechnikov kernel implemented in this package with the `Epanechnikov` function. The null model f_0 is specified through the `dist` argument with parameters passed through the `p1` and `p2` arguments. The test is implemented in conjunction with `hopt.cutoff` function which provides the value of h needed for calculation of $S_n(h)$ and the critical value used to determine acceptance or rejection of the null hypothesis. See the example below for an application to a real world dataset.

The test statistic can be thought as a descritized version of the bias corrected test statistic in page 380 of Li and Racine (2007), using the nonparametric empirical density estimate in place of the classical kernel density estimate there.

Value

A vector with the value of the test statistic as well as the bandwidth used for its calculation

Author(s)

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

References

Li and Racine, Nonparametric Econometrics: Theory and Practice, Princeton University Press, (2007)

See Also

hopt.cutoff

Examples

```
library(fGarch)
library(kedd)
data(EuStockMarkets)
DAX <- as.ts(EuStockMarkets[,"DAX"])
dax <-  diff(log(DAX))#[,"DAX"]

# Fit a GARCH(1,1) model to dax returns:
lll<-garchFit(~ garch(1,1), data = as.ts(dax), trace = FALSE, cond.dist ="std")
# define the model innovations, to be used as input to the test statistic
xin<-lll@residuals /lll@sigma.t
# exclude smallest value - only for uniform presentation of results (this step can be excluded)
xin = xin[xin!= min(xin)]

#inputs for the test statistic:
#kernel function to use in implementing the statistic and functional estimates for optimal h:
kfun<-"epanechnikov"
a.sig<-0.05 #define the significance level

Nulldist<-"normal" #null hypothesis is that the innovations are normaly distributed


p1<-mean(xin) # mean of the
p2<- sd(xin) #

cutoff<-hopt.cutoff(xin,   Nulldist, kfun, p1, p2, a.sig ) # power optimal bandwidth


TestStatistic<-S.n(xin, cutoff[1], 1/3, Nulldist, p1, p2)


cat("L2 test statistic value:", TestStatistic[1],  " critical value :", cutoff[2], "\n")
 # normality is rejected
```

Index